Fermilab

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Large Scale Management of Physicist's Personal Analysis Data without Employing User and Group Quotas

A. Norman, M. Diesburg, M. Gheith, R. Illingworth, M. Mengel

Fermilab, Scientific Computing Division
Scientific Data Management

Data Sizes

- The FNAL Neutrino & Muon program are generating a large amount of data both in terms of raw bytes and total file counts

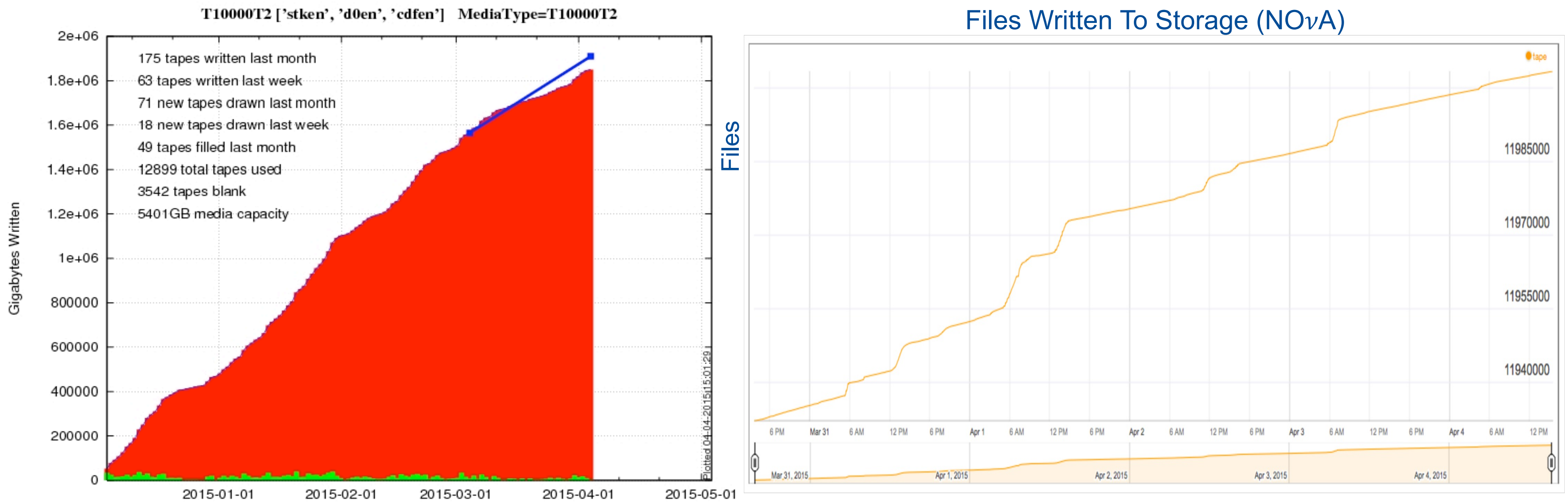


FIG. 1 NOνA data and file volumes corresponding to first 12 months of physics operation and preparation for first analysis results. Total accumulated data to date 1.6 PB and over 12M files. Totals represent only "official" datasets.

The Problem of Storage

- The more data you have the harder it is to organize store and retrieve it effectively
- Essentially 3 Domains:

Conventional Random Access (Big Disk)

Properties

- Local or Centralized Disk
- Standard DAS or NAS
- Normally POSIX
- Scales poorly (size and load)
- Availability/Reliability
- High Cost
- Easy to Use
- Low latency
- Intermediate Throughput

Storage Elements, Object Stores & Cache

Properties

- Centralized or Distributed
- May be exposed as NAS or SAN
- Typically non-POSIX
- Can scale capacity/load
- Redundancy + High Availability
- Intermediate cost
- Difficult for physicists to use directly
- Low latency
- High Throughput

Archival Storage (Tape)

Properties

- Centralized Facility (dedicated infrastructure)
- May not be exposed at all
- non-POSIX
- Capacity scales easily
- Concurrent load does not scale well
- "Archival"
- Lowest Cost
- **VERY DIFFICULT** for physicists to use
- High latency
- Low throughput

The Problem of Storage

- The more data you have the harder it is to organize store and retrieve it effectively
- Essentially 3 Domains:

Conventional Random Access (Big Disk)

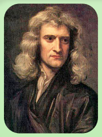
Properties

- Local or Centralized Disk

- Standard DAS or NAS
- Normally POSIX

- Scales poorly (size and load)
- Availability/Reliability
- High Cost

- Easy to Use
- Low latency
- Intermediate Throughput



Physicist Wants His Data Here

Storage Elements, Object Stores & Cache

Properties

- Centralized or Distributed

- May be exposed as NAS or SAN
- Typically non-POSIX

- Can scale capacity/load
- Redundancy + High Availability
- Intermediate cost

- Difficult for physicists to use directly
- Low latency
- High Throughput

Analysis Wants Data Here

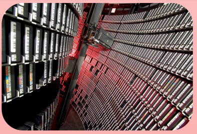
Archival Storage (Tape)

Properties

- Centralized Facility (dedicated infrastructure)
- May not be exposed at all
- non-POSIX

- Capacity scales easily
- Concurrent load does not scale well
- “Archival”
- Lowest Cost

- **VERY DIFFICULT** for physicists to use
- High latency
- Low throughput



We can afford Data Here

The Problem of Storage

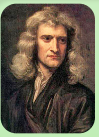
- The more data you have the harder it is to organize store and retrieve it effectively
- Essentially 3 Domains:

Successful in moving production here

Conventional Random Access (Big Disk)

Properties

- Local or Centralized Disk
- Standard DAS or NAS
- Normally POSIX
- Scales poorly (size and load)
- Availability/Reliability
- High Cost
- Easy to Use
- Low latency
- Intermediate Throughput



Physicist Wants His Data Here

Storage Elements, Object Stores & Cache

Properties

- Centralized or
- May be expose
- Typically non-F
- Can scale capa
- Redundancy +
- Intermediate c
- Difficult for physicists to use directly
- Low latency
- High Throughput

Analysis Wants Data Here

Archival Storage (Tape)

Properties

cale well

- VERY DIFFICULT for physicists to use
- High latency
- Low throughput

We can afford Data Here

FIFE Suite:

Data Handling Tools
SAM, IFDH, F-FTS

Analysis Tools
art framework, jobsub wrappers

Data Composition

- The data currently in managed storage is “production”
- Well understood both in size and file counts
- Robust, mature, complete tools chains to work with the data

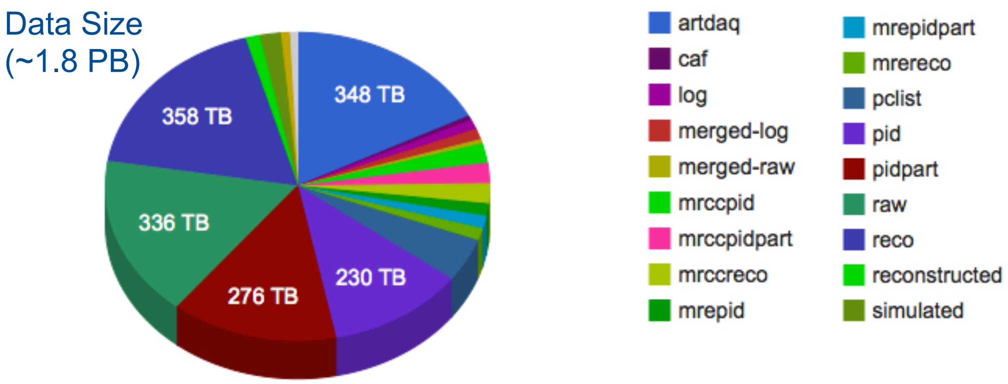


FIG 1. Storage usage by data type for NOνA first analysis data sets. Storage is dominated by the production chain raw→calib→reco→pid1→pid2

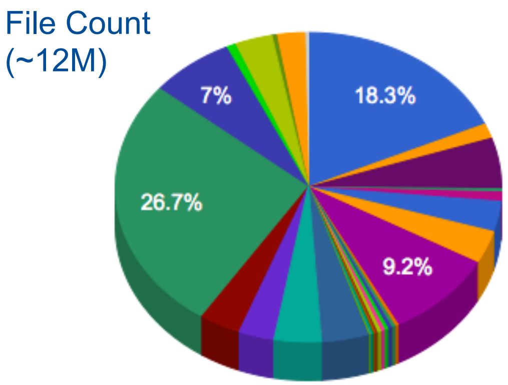


FIG 2. File counts by data type for NOνA first analysis data sets. File counts are dominated by the raw data and calibration stages

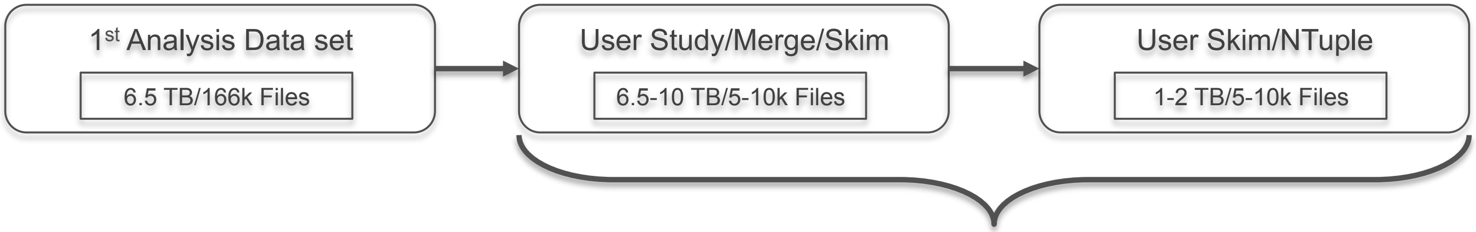
Can this be expanded to include user level skims & analysis Ntuples?

User Skims & Ntuples

- Example: NOνA First Analysis—Far Detector Beam data

Type	Events (Spills)	Files	Size
Official 1st Analysis Dataset	14,308,325	166,629	6.51 TB

- This is the skimmed down signal set for ≈7 months of data. (the background set is 10x the size after a 10x reduction)
- This is the starting point for most users to do analysis


1st Analysis Data set

6.5 TB/166k Files

User Study/Merge/Skim

6.5-10 TB/5-10k Files

User Skim/NTuple

1-2 TB/5-10k Files

Each Physicist Generates 7-12 TB of data spread over 10-20k Files (per study!)

User Skims & NTuples

- The problem is that there isn't just 1 physicist looking at data on each experiment
- An experiment like NOνA has over 100 unique physicists, postdocs, students who HAVE active analysis areas on the central project disks & dCache scratch areas!
  - Half of these already have dedicated skims/ntuples etc…
- This would mean:

0.3-0.6 PB of studies/skims/ntuples spread over 1-2M files

This is what we actually see 6 months into analysis

Data Management

- How do you manage all this data?

You don't. You can't.

There are too many files in too many locations and there is no record of what it is.

- Quotas Don't work

They limit what a person can store, but don't organize the information

When Quotas are reached there isn't a "cleanup" mechanism

Quotas require humans to manage and adjust them

Data Management Without Quotas

The key requirements for physicist's data management are:

1. **IT MUST BE TRIVIAL TO USE**
2. **MUST INTEGRATE WITH ANALYSIS TOOLS**
3. **MUST ALLOW FOR CLEANUP AND ARCHIVING OF DATA**

The model we adopted was a "DATA CATALOG LITE"

- integrates with the standard analysis tools and frameworks
- tools to provide common **task based** functions
  - operate transparently against archival, cache, distributed or traditional storage.
- *removes the "**file**" from how the physicist operates*

SAM for the Physicist

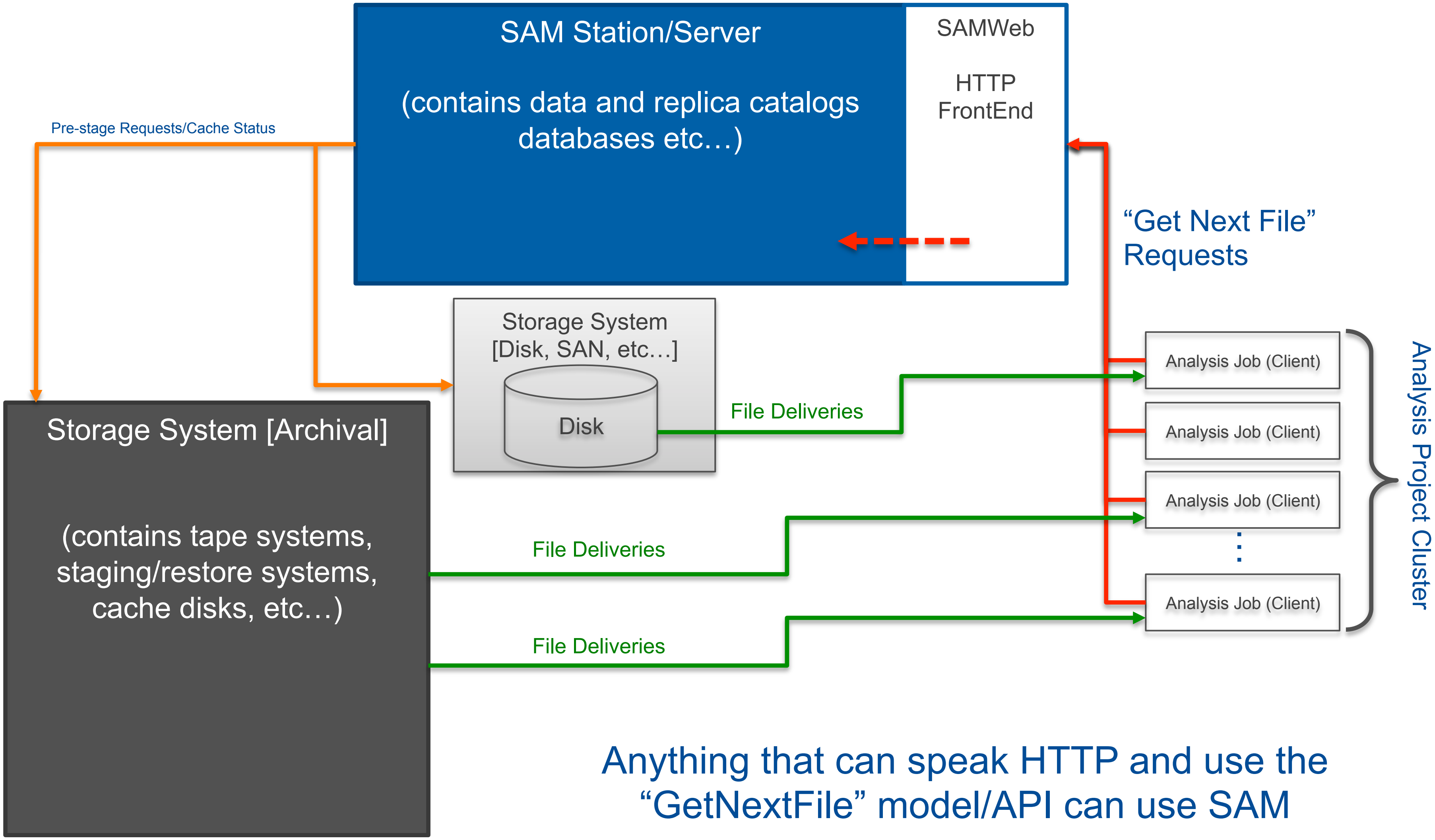
- The full featured SAM (Sequential Access w/ Metadata) system integrates a Data and Replica catalog with data movement and analysis project scheduling
  - It is designed for optimized file delivery from archival storage
  - Heavily used by Fermilab experiments since Run II.
  - Mainly a “production” tool due to older architecture requirements

- SAMWeb & SAMLITE relax the architecture requirements on data to make it possible to provide an “EZ” interface that analysis users can use
  - EXAMPLE: DOES NOT REQUIRE USER SUPPLIED METADATA TO REGISTER FILES

Central Concept: The Dataset

- Central to SAM is the Dataset
  - A Dataset is a collection of files that “belong” together based on some meta information
    - For production tasks these are complicated relational queries related to the actual physics
    - But these can be as simple as:
“They belong together because I said so”
  - Analysis is run against a Dataset
    - Jobs have no a priori knowledge of which files they get.
      - Ordering does not matter. Files are delivered at run time.
    - SAM optimizes the delivery order
(based on availability & infrastructure)

Analysis w/ SAM



Anything that can speak HTTP and use the "GetNextFile" model/API can use SAM

SAM for Users Tools

- First tool is the “Add” a dataset tool
  - Associates a group of files as a dataset, “because I said so” (limited metadata)

```
sam_add_dataset  -n <dataset name> -d <directory path>
```

  - All files in the directory (and optionally subdirs) are:
    - Registered with SAM
    - Replica information is recorded
    - Name collisions are prevented (in namespace)
    - Associated and made into a usable dataset.
  - Eliminates the confusion of dealing with individual files
  - Scales appropriately (i.e. 10k’s of files are fine)

SAM for Users Tools

- Additional simplified user tool set operates on the “dataset” as a unit to make common tasks easy

Simplified User “task” Functions

clone/unclone	Create/Copy/Remove replicas to other managed storage
pin	Extends TTL of the data on volatile storage
validate	Validates a replica
modify_metadata	Update or add meta information
retire	Delete the dataset and/or associated files

- All other functionality is provided by full SAM system (i.e. catalogs, data transport, etc…)

How does this eliminate the need for quotas?

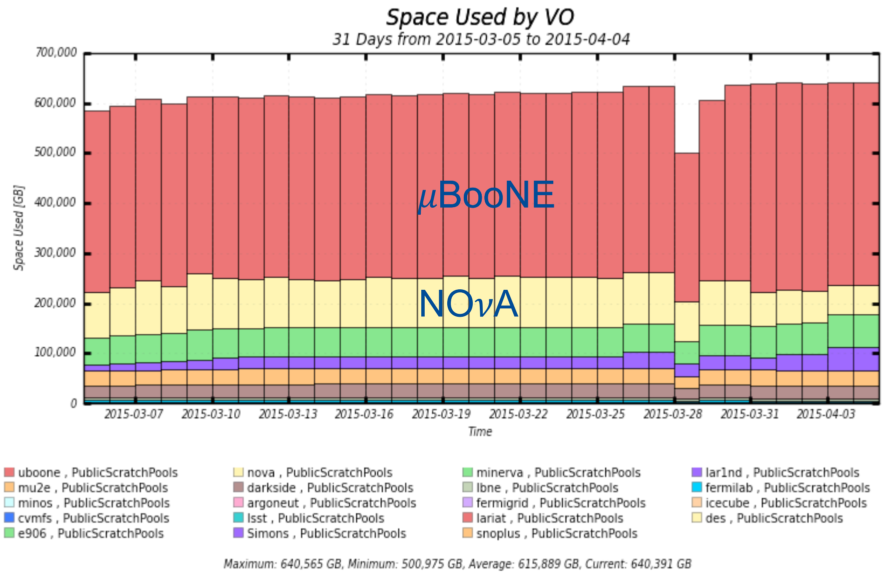
- Provides a reduction in complexity
  - Instead of millions of individual files, physicists deal with a handful of dataset "names"
- Provides operational capability
  - Insulates physicists from having to understand how more complicated storage system operate.
  - They just need to know their "dataset" name to analyze it
- Provides automated "cleanup" functions:
  - Data movement, Archiving, Removal
  - Without the need to know "where" things are

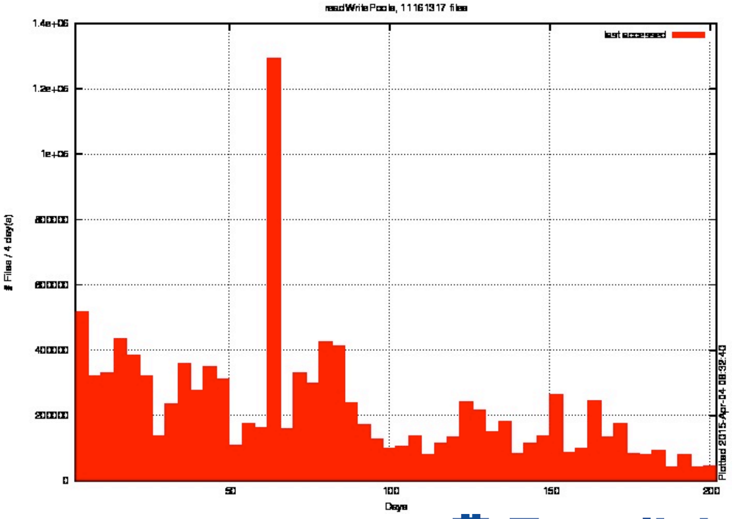
Example: Volatile dCache Storage

- Volatile storage at FNAL is a large (0.64 PB) dCache pool group which does NOT have tape backing
- Shared between experiments

- The pool uses and Least Recently Used (LRU) cache algorithm to manage/expire data
- The average TTL for files is 60 days

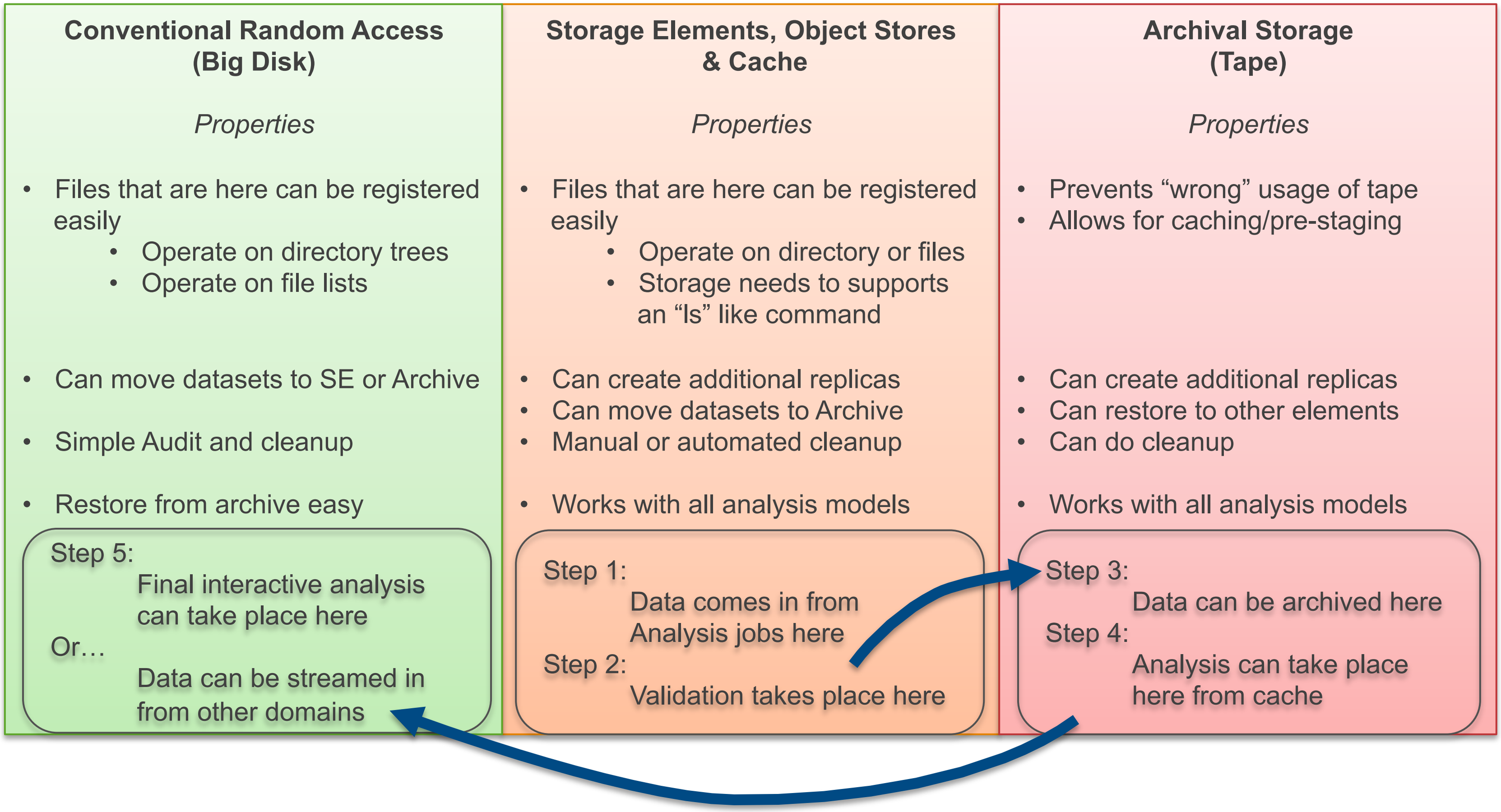
- Designed to home “temporary” analysis files for validation, studies, etc…
  - This is where you want to operate for performance reasons
  - But you are worried about your files disappearing





Quota Less Management

- Now the physicist is capable of operating in all three domains



Conventional Random Access (Big Disk)

Properties

- Files that are here can be registered easily
  - Operate on directory trees
  - Operate on file lists
- Can move datasets to SE or Archive
- Simple Audit and cleanup
- Restore from archive easy

Step 5:
Final interactive analysis can take place here
Or…
Data can be streamed in from other domains

Storage Elements, Object Stores & Cache

Properties

- Files that are here can be registered easily
  - Operate on directory or files
  - Storage needs to supports an “ls” like command
- Can create additional replicas
- Can move datasets to Archive
- Manual or automated cleanup
- Works with all analysis models

Step 1:
Data comes in from Analysis jobs here
Step 2:
Validation takes place here

Archival Storage (Tape)

Properties

- Prevents “wrong” usage of tape
- Allows for caching/pre-staging
- Can create additional replicas
- Can restore to other elements
- Can do cleanup
- Works with all analysis models

Step 3:
Data can be archived here
Step 4:
Analysis can take place here from cache

Conclusions: It works….

- This actually works for:
  - Standard "analysis jobs"
    - Official framework
      (100k's of files, many TB data)
  - Analysis/Study Skims
    - Custom analysis frameworks
      (10k's of files, < TB)
  - Analysis NTuples
    - Interactive ROOT sessions
      (chained trees w/ streaming via xrootd)

- First week after initial released had 260,897 user files registered and analysis underway

Conclusions

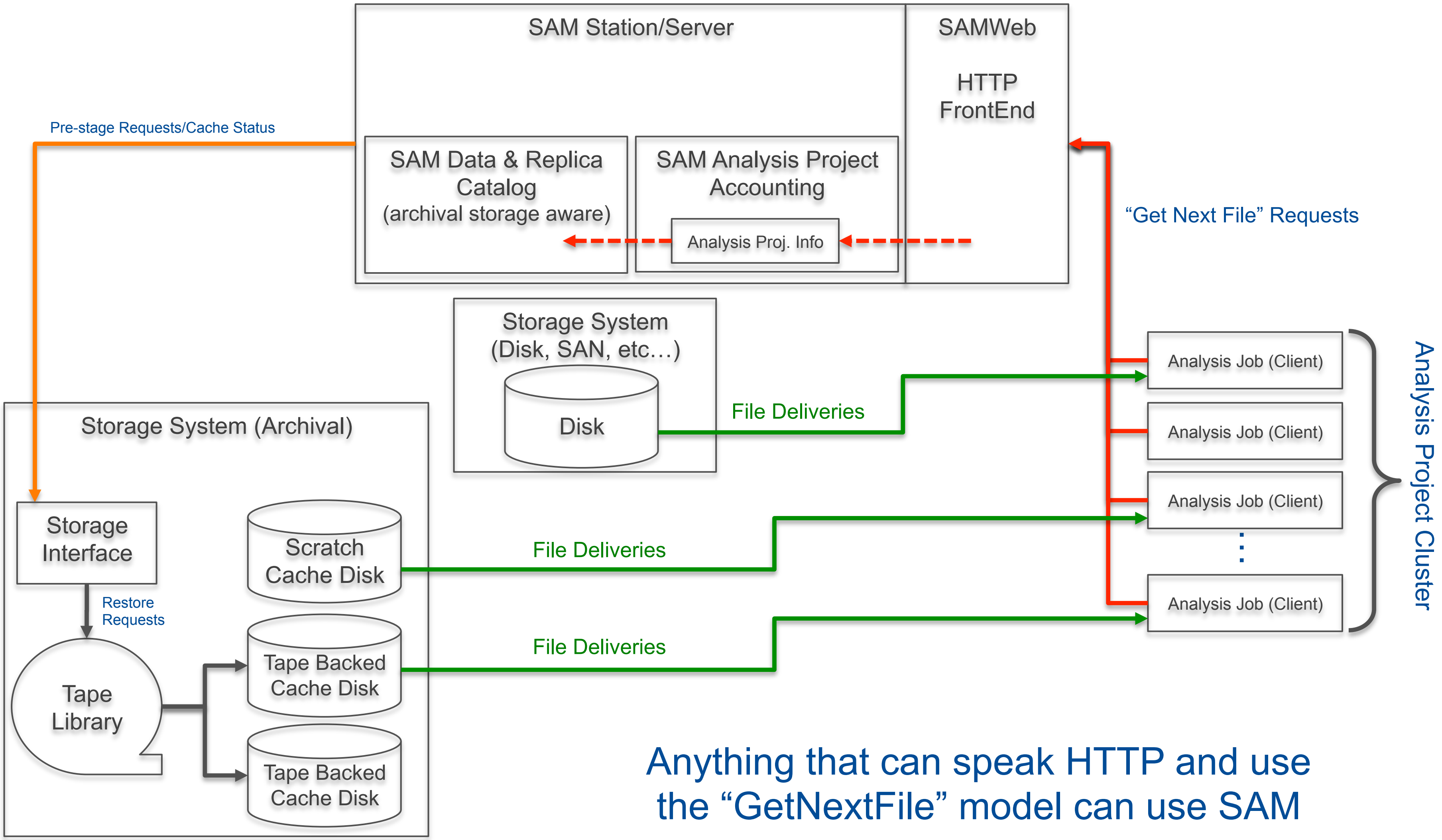
- We have created set of end user -- “physicist” tools which are able to provide full featured data management

- The tools leveraged & expanded the SAM data handling system
  - Full tool set required < 1 week of develop prior to first release

- The tools were designed to ease the use of distributed and archival storage system

- Wide spread adoption by users on the experiments

Analysis w/ SAM



Anything that can speak HTTP and use the "GetNextFile" model can use SAM